Title: How Attackers Abuse Computing Systems

Author(s): Kennel, David A.

Intended for: Presentation for ISTI summer cluster program students.

Issued: 2015-06-18

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer,is operated by the Los Alamos National Security, LLC for the National NuclearSecurity Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Departmentof Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

How Attackers Abuse Computing Systems

The following story is true, only the names have been changed to protect the innocent. . .

9:08 AM

- J. Random User, an accountant with Technocorp receives an email message with an attached document from BigBoss@gmail.com. The email subject line and body reference an important current project and the attached spreadsheet claims to contain a quote for the project.
- Gmail is not normally used for corporate email but a number of employees have private email addresses.
- The boss is on travel and the project references are all correct.

9:15 AM

- J. Random User's system contacts a domestic webserver over port 443. This appears to be normal HTTPS traffic and Technocorp's overworked IT staff ignores it.
- J. Random User's system downloads and installs a rootkit and several other utilities.

9:45 AM

- Simon, one of Technocorp's System Administrators, logs onto J. Random User's machine to fix a configuration issue. Simon uses his domain administrator credentials.

10:30 AM

- Using the “pass the hash” technique attackers begin moving through the network using Simon's credentials.
- By 11 AM the attackers own Technocorp's AD servers.
- Within 48 hours the attackers own Technocorp's SCCM servers and all major file servers.
- Within 48 hours the attackers begin siphoning critical intellectual property out of Technocorp.
- The attack does not trigger anti-virus or IDS/IPS defenses.
- Technocorp will only find out about the attack 3 months later when they are told by the FBI that attackers are in the Technocorp network.

How Attackers Abuse Computing Systems

What Security Is. . .

- Implementing controls and configurations to protect the Confidentiality, Integrity and Availability of a computing system and it's data.
- Security is not about 100% defenses.
  - One of the reasons security is hard: We must defend every possible avenue of attack. The opponent only has to find one weakness.
- Goal is to delay an attacker long enough to catch or to make the attacker move on to an easier target.
  - When you find yourself in the company of a halfling and an angry dragon, remember you don't have to outrun the dragon. . . you only have to outrun the halfling.

The Enemy

- Script Kiddies
- Disorganized Crime
- Organized Crime
- Advanced Persistent Threat
- Your Users
  - “When confronted with the prospect of being fired tomorrow and ethics going out the door, 71 per cent surveyed declared they would definitely take company data with them to their next employer.” (http://www.scmagazineuk.com/IT-workers-would-steal-data-in-the-event-of-a-redundancy-threat/article/122301/)
- Your Co-Workers

The Enemy

- **YOU**
  - Microsoft Tech Net **Security in Operation (4/4): Managing Security:** “Zone-H found that the single largest factor in successful attacks was administrator misconfiguration, cited in 33 percent of the attacks; the second largest factor was unpatched vulnerabilities, cited in 25 percent.”

The Threat

- **Advanced Persistent Threat:**
  - Deeply resourced.
  - Target is IP not financial data
  - Will conduct thorough research on targets
  - Crafted attacks, typically does not use shotgun attacks
  - Usually has nation state sponsorship
  - Aurora – Claimed by Google to be an APT attack
  - GhostNet, Shadow Network – investigated by Shadowserver Foundation and the Canadian Information Warfare Monitor

The Threat

- **Advanced Persistent Threat attack stages:**
  - Reconnaissance
  - Network Intrusion (spear phishing)
  - Backdoor
  - Grab Credentials
  - Install utilities
  - Privilege escalation, lateral movement and exfiltration
  - Maintaining persistence

The Threat

- Increasingly sophisticated polymorphic, packed malcode with self-update capability.
- Increasing use of encryption.
- Increasing code quality.
- Increasing complexity of computing environments.
- The digital domain is more interesting and rewarding for criminals and espionage every year.

PARANOIA
Sometimes paranoia's just having all the facts. William S. Burroughs

How Attackers Abuse Computing Systems

Social Engineering

"Because there is no patch for human stupidity"

To Catch A Phish

LAGOS, NIGERIA.

ATTENTION: THE PRESIDENT/CEO

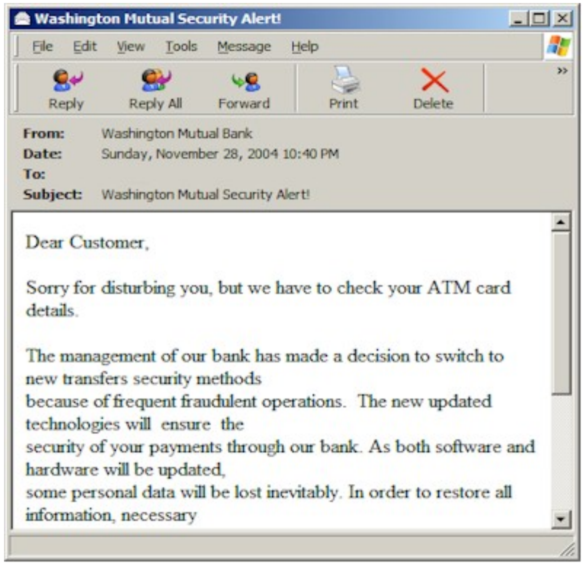
DEAR SIR,

CONFIDENTIAL BUSINESS PROPOSAL

HAVING CONSULTED WITH MY COLLEAGUES AND BASED ON THE INFORMATION GATHERED FROM THE NIGERIAN CHAMBERS OF COMMERCE AND INDUSTRY, I HAVE THE PRIVILEGE TO REQUEST FOR YOUR ASSISTANCE TO TRANSFER THE SUM OF $47,500,000.00 (FORTY SEVEN MILLION, FIVE HUNDRED THOUSAND UNITED STATES DOLLARS) INTO YOUR ACCOUNTS.

To Catch a Phish

- **Bank Fraud**
- **Spear Phishing**
  - March 2009 researchers investigating intrusions in the Tibetan exile centers in India discover a global network of compromised machines they name “Ghost Net”. Spear phishing attacks played a key role in the intrusion.



From: Washington Mutual Bank
Date: Sunday, November 28, 2004 10:40 PM
To:
Subject: Washington Mutual Security Alert!

Dear Customer,

Sorry for disturbing you, but we have to check your ATM card details.

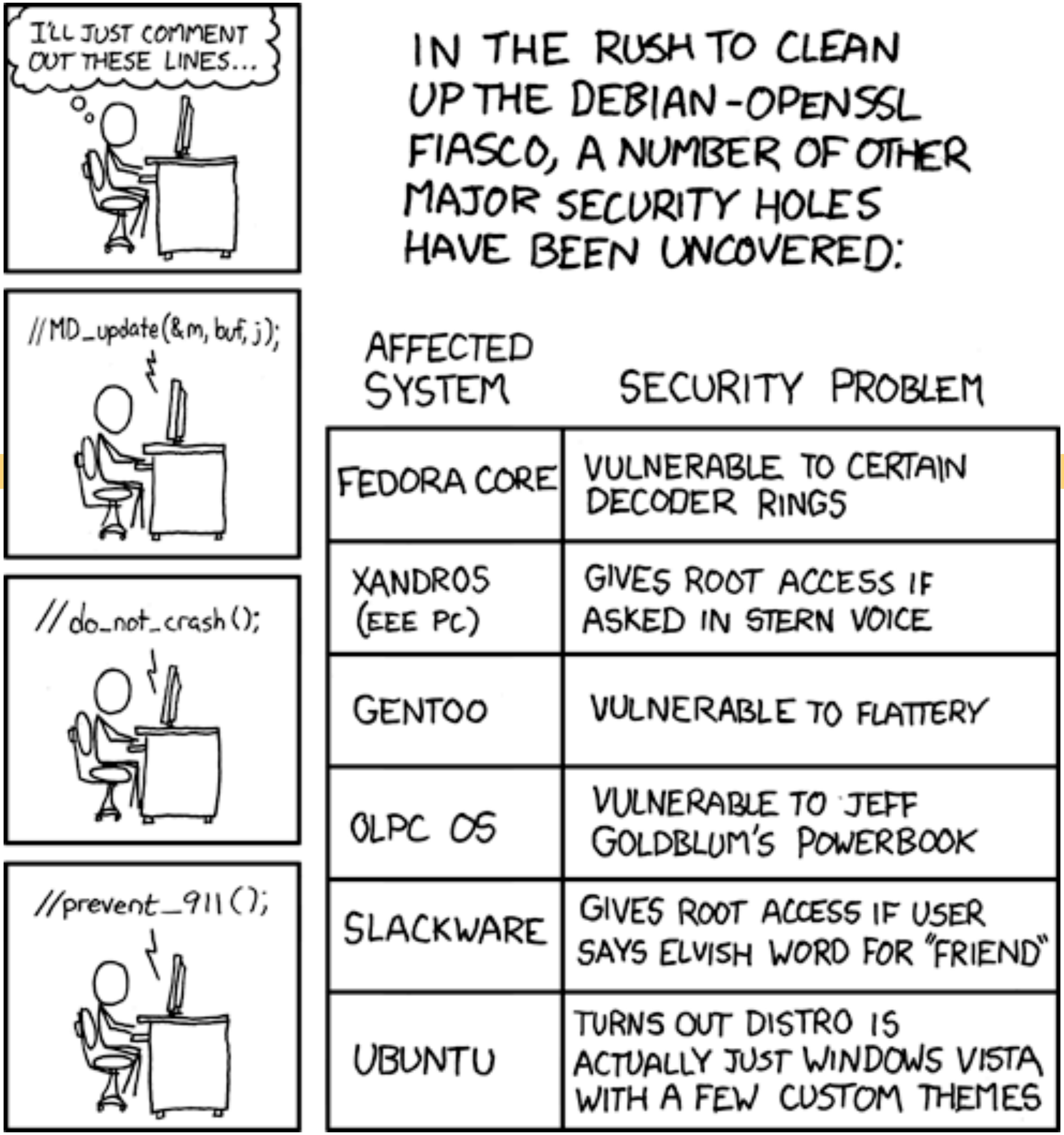
The management of our bank has made a decision to switch to
new transfers security methods
because of frequent fraudulent operations. The new updated
technologies will ensure the
security of your payments through our bank. As both software and
hardware will be updated,
some personal data will be lost inevitably. In order to restore all
information, necessary

Other Attacks Social and Otherwise

- Impersonation of authorized or service personnel
- Tailgating
- Dumpster Diving
- Shoulder Surfing
- Chocolate for Passwords? (70%?!?!)
- Open source intelligence gathering

How Attackers Abuse Computing Systems

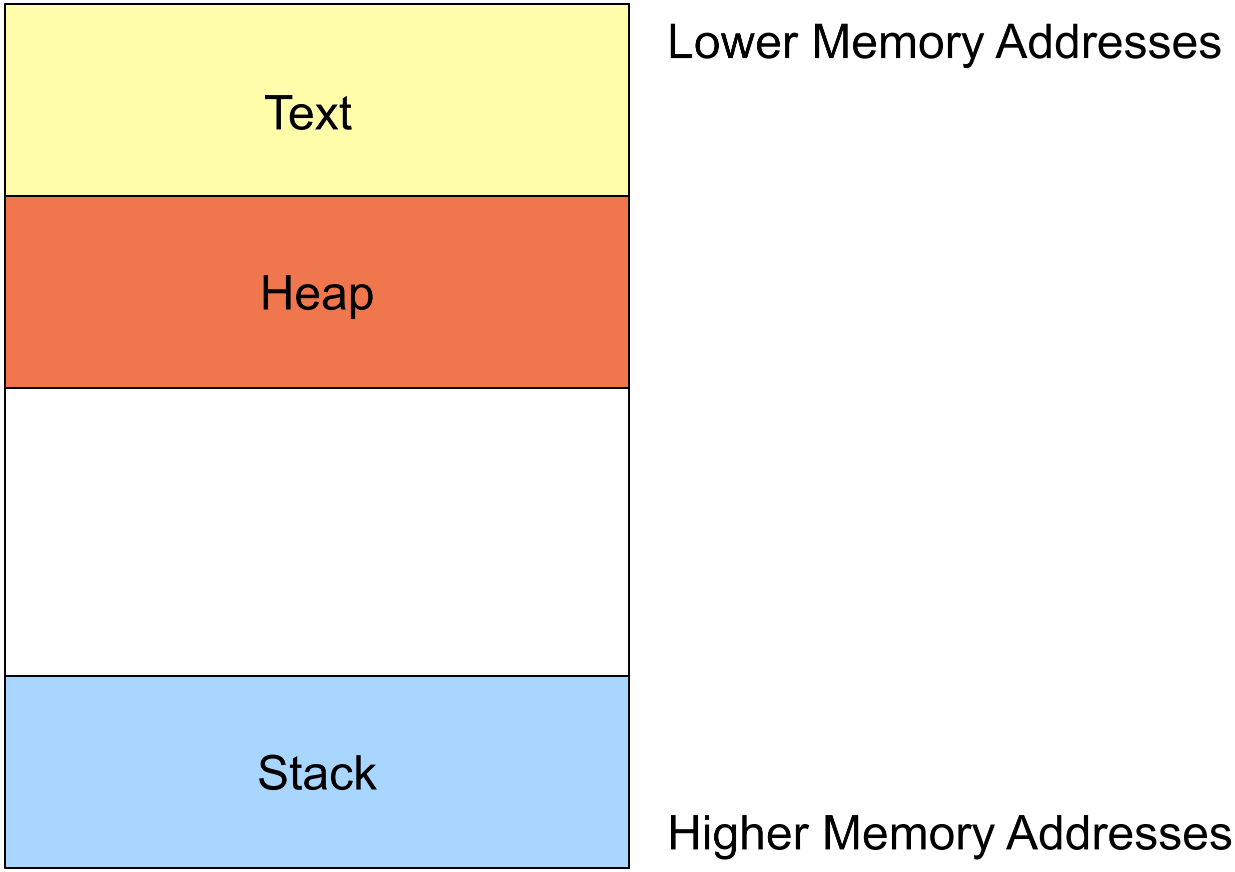
Abusing Vulnerable Software



IN THE RUSH TO CLEAN UP THE DEBIAN-OPENSSL FIASCO, A NUMBER OF OTHER MAJOR SECURITY HOLES HAVE BEEN UNCOVERED:

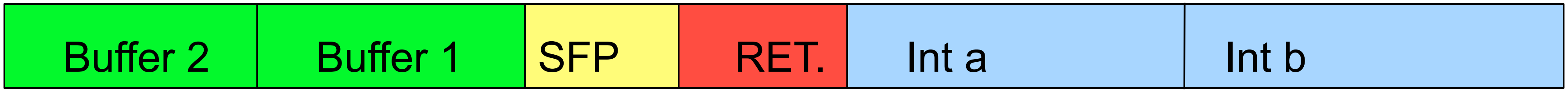
AFFECTED SYSTEM	SECURITY PROBLEM
FEDORA CORE	VULNERABLE TO CERTAIN DECODER RINGS
XANDROS (EEE PC)	GIVES ROOT ACCESS IF ASKED IN STERN VOICE
GENTOO	VULNERABLE TO FLATTERY
OLPC OS	VULNERABLE TO JEFF GOLDBLUM'S POWERBOOK
SLACKWARE	GIVES ROOT ACCESS IF USER SAYS ELVISH WORD FOR "FRIEND"
UBUNTU	TURNS OUT DISTRO IS ACTUALLY JUST WINDOWS VISTA WITH A FEW CUSTOM THEMES

Process Memory Organization



Inside the Stack

```
void function(int a, int b) {
   char buffer[5];
   char buffer2[10];
}
void main() {
   function(1,2);
}
```



Buffer 2 | Buffer 1 | SFP | RET. | Int a | Int b

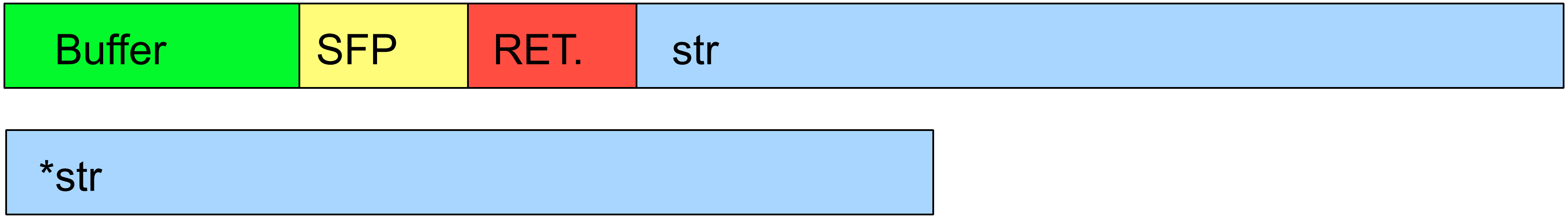
Top of stack

Bottom of stack

Buffer Overflow

```
void function(char *str) {
   char buffer[16];
   strcpy(buffer, str);
}
void main() {
   char big_string[256];
   int i;
   for( i = 0, i < 255; i++)
       large_string[i] = 'A';
   function(large_string);
}
```

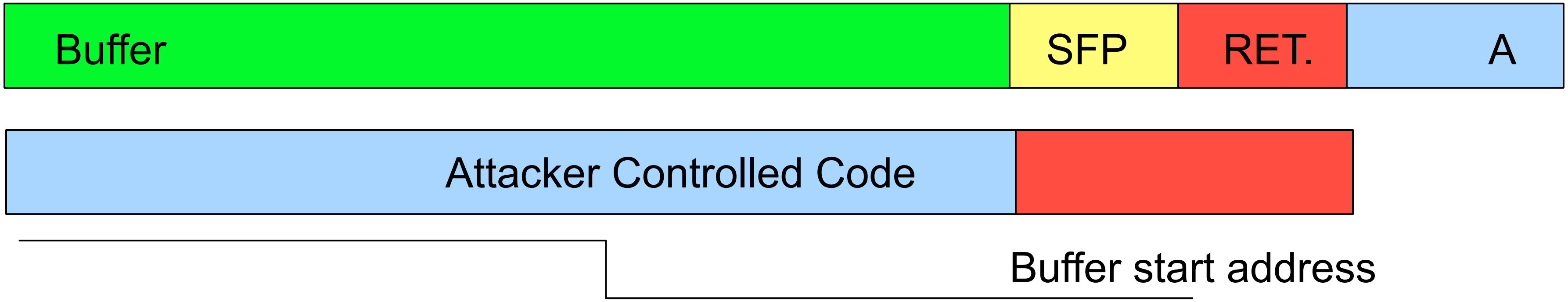
Buffer Overflow cont.



A = 0x41

New return address is 0x41414141

Running Arbitrary Code



What Does “Arbitrary Code” Mean

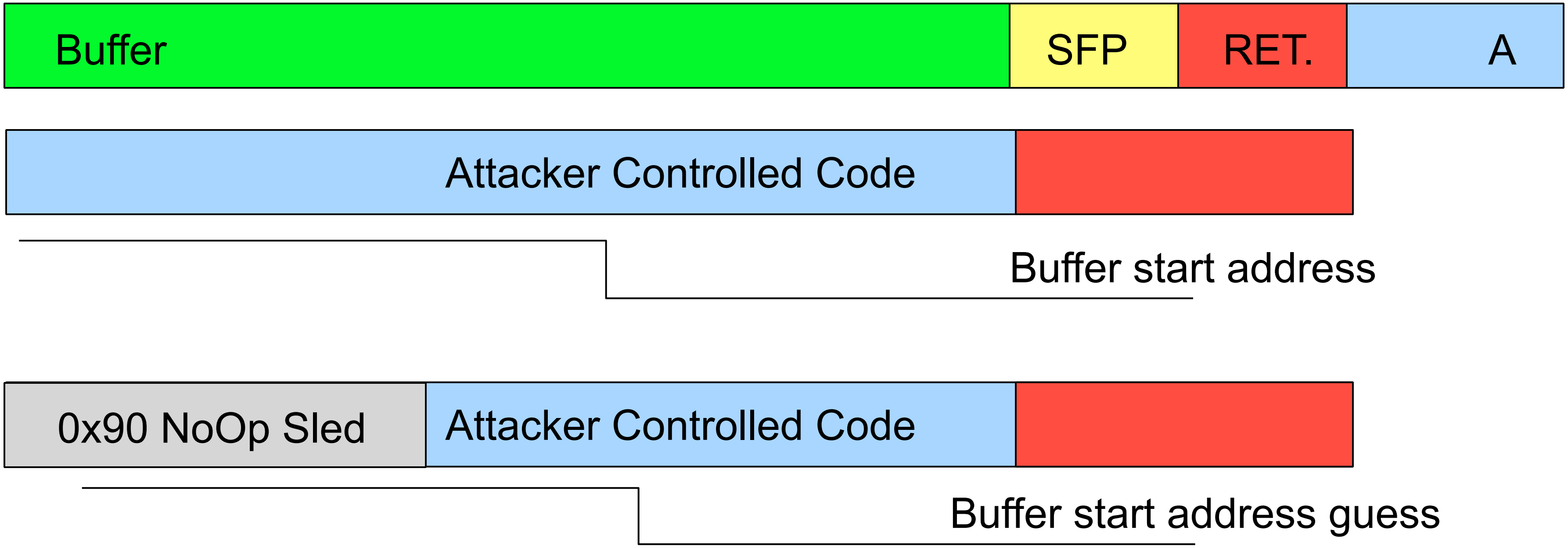
```
char shellcode[] =

"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"

"\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"

"\x80\xe8\xdc\xff\xff\xff/bin/sh";
```

Real World Issues



Buffer

SFP

RET.

A

Attacker Controlled Code

Buffer start address

0x90 NoOp Sled

Attacker Controlled Code

Buffer start address guess

Commonly Exploited Vulnerabilities

- **Programming errors:**
  - Buffer overflow
  - Format string attacks
  - SQL injection
  - Command injection
  - Cross Site Request Forgery
  - Dangling or Wild Pointers
  - Time of check, time of use (TOCTOU)
  - Many, many, many, more

Commonly Exploited Vulnerabilities

- **Microsoft SDLC Banned Function Calls**
  - strcpy, wcscpy, _tcscpy, _mbscpy, StrCpy, StrCpyA, StrCpyW, lstrcpy, lstrcpyA, lstrcpyW, strcpyA, strcpyW, _tccpy, _mbccpy, strcpy, wcscpy, _tcscpy, _mbscpy, StrCpy, StrCpyA, StrCpyW, lstrcpy, lstrcpyA, lstrcpyW, strcpyA, strcpyW, _tccpy, _mbccpy, wnsprintf, wnsprintfA, wnsprintfW, sprintfW, sprintfA, wsprintf, wsprintfW, wsprintfA, sprintf, swprintf, _stprintf, _snwprintf, _snprintf, _sntprintf, nsprintf, wvsprintf, wvsprintfA, wvsprintfW, vsprintf, _vstprintf, vswprintf, _vsnprintf, _vsnwprintf, _vsntprintf, wvnsprintf, wvnsprintfA, wvnsprintfW, strncpy, wcsncpy, _tcsncpy, _mbsncpy, _mbsnbcpy, StrCpyN, StrCpyNA, StrCpyNW, StrNCpy, strcpynA, StrNCpyA, StrNCpyW, lstrcpyn, lstrcpynA, lstrcpynW, _fstrncpy, strncat, wcsncat, _tcsncat, _mbsncat, _mbsnbcat, StrCatN, StrCatNA, StrCatNW, StrNCat, StrNCatA, StrNCatW, lstrncat, lstrcatnA, lstrcatnW, lstrcatn, _fstrncat . . .

Buffer Overflow Defenses

- **Address space layout randomization**
- **Data execution prevention**
- **Compiler inserted “canary” values**
- **Defensive programming**
  - Automated code analysis
  - Code review

How Attackers Abuse Computing Systems

Other Examples of System Abuse

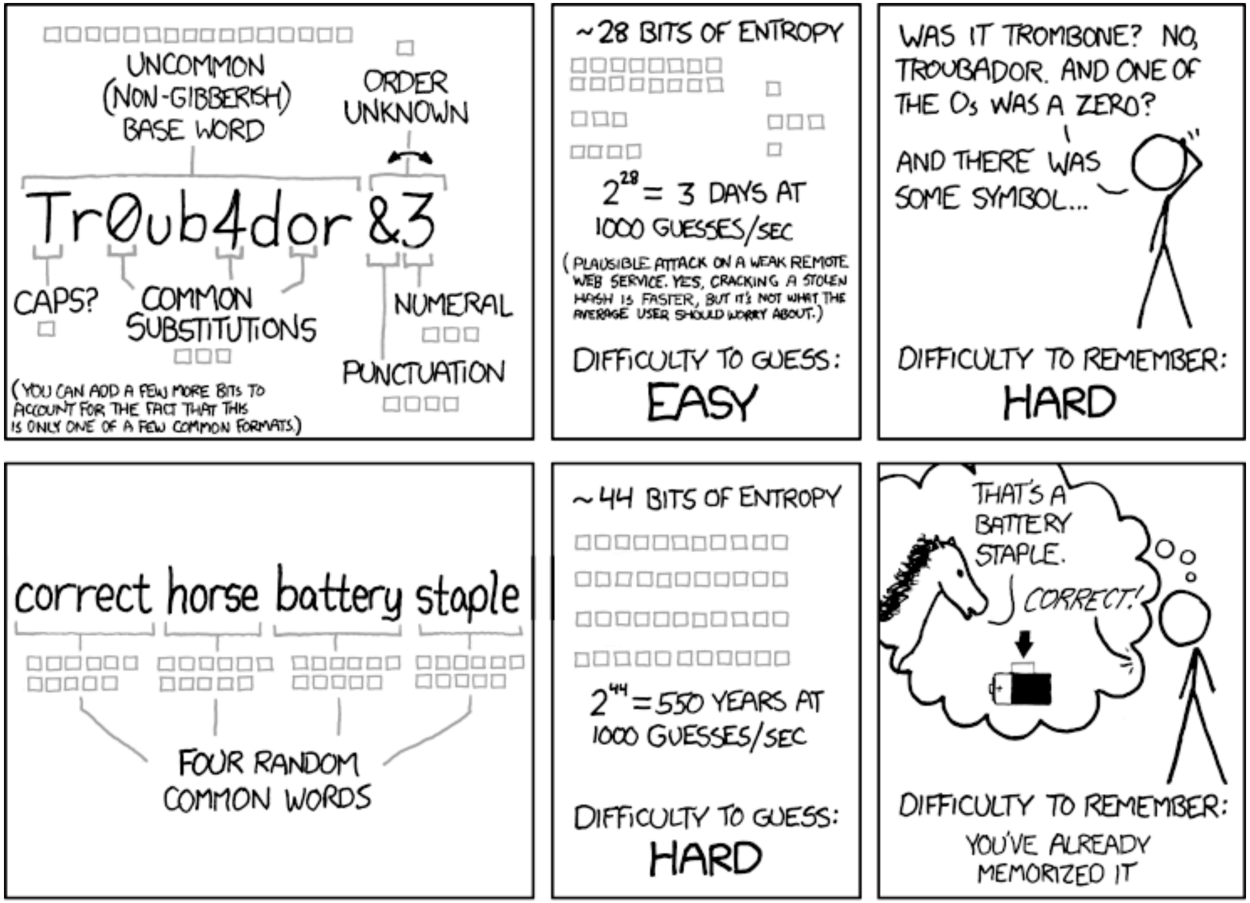
Resource Exhaustion Attacks

- **:(){ :|:& };:**
  - Linux Bash/KSH fork bomb
- **%0|%0**
  - Windows variant
- **fork while 1**
  - Perl
- **Mitigating fork bomb attacks:**
  - **Resource controls: Ulimit**
  - **/etc/security/limits.conf**
- **Denial of service:**
  - **Smurf (broadcast ping), Fraggle (broadcast UDP), echo/chargen loop**
  - **NIS – finger user@host.**

Weak Passwords

- Weak passwords are one of the top 5 ways that attackers get system access.
- Password crackers and rainbow tables on modern hardware are capable of doing thousands of attempts per second.
- Password cracking dictionaries, permutation checkers and rainbow tables are now so sophisticated that they can crack passwords more complex than most people can remember.
- At this time there is a sophisticated, automated password guessing attack against SSH on the internet.

Weak Passwords


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor &3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Yet More Abuse – Weak Authentication

- .rhosts (specifies users/hosts that can rlogin/rsh without a password)!
  - + + = Any user on any host.
  - Common target for file overwrite attacks.
  - Tree of additional machines to access.
- Xhost authentication:
  - Xhost +

Yet More Abuse – Weak Authentication

- RPC authentication:
  - Service registration.
  - UID trust issue.
- NFS:
  - Trusts UIDs.
  - Packet injection into NFS sessions.

File System Abuse – Exploiting Race Conditions

- Programs writing to directories with unsafe permissions.
- Compounded by programs which choose easy to predict file names.
- Attack vectors:
  - Manipulate unexpected files (DOS, System Access).
  - Execute attacker controlled code.

Example – 1992 SunOS /bin/mail Exploit

- /bin/mail called by Sendmail to deliver mail to user mail spools at /var/spool/mail
- /bin/mail checks destination file to see if it is a symlink and then opens the file for writing. Not an atomic action.
- Between the check and the open the attacker creates a symlink to /.rhosts. Then an email message is sent to root containing "+ +".
- http://www.outpost9.com/exploits/mail.8lg

Yet More Abuse

- Giving away too much info:
  - Chatty services that have too much info in banners and connection messages.
  - Reduces time needed to determine exploitability.
- NIS – ypcat passwd will show passwords even though /etc/shadow exists.

More Abuse

- Example of attackers using non-linear thinking:
  - Unix systems moved password hashes to a non world readable location.
  - Fingerd is a common service on older Unix systems.
  - Old versions of Fingerd run as root.
  - Attacker links own .plan to /etc/shadow and then runs finger.
  - Attacker proceeds to crack passwords.

How Attackers Abuse Computing Systems

Rootkits

Rootkits

- The attackers goal is usually to maintain control of a system for some other purpose.
- Rootkits are groups of tools that enable the attacker to maintain access.
- Common rootkit behavior:
    - Process hiding
    - File hiding
    - Remote access/command channel

Types of Rootkits

Traditional:

- Trojan replacements for standard system utilities: ls, ps, netstat, etc.
- Checksumming defenses detect this type of rootkit easily.

Kernel:

- Attach directly to the kernel.
- Are able to intercept utilities at the system call level.
- Much harder to detect.

Detecting Rootkits

- Analyze system using an OS CD-ROM.
- All detection/analysis tools must be trustworthy.
- If the system is rooted - reinstall.

How To Defend Computing Systems

UNCLASSIFIED

Top 20 Critical Security Controls

1. Inventory of Devices
2. Inventory of Software
3. Secure Configurations for Computers
4. Continuous Vulnerability Assessment and Remediation
5. Malware Defenses
6. Application Software Security
7. Wireless Device Control
8. Data Recovery Capability
9. Security Skills and Training
10. Secure Configurations for Network Devices
11. Control of Network Ports & Protocols
12. Controlled Use of Admin Privileges
13. Boundary Defenses
14. Monitoring Logs
15. Need to Know
16. Account Monitoring and Control
17. Data Loss Prevention
18. Incident Response Capability
19. Secure Network Engineering
20. Penetration Tests

Principles of Defense

- Secure Configuration
- Patch Early Patch Often
- Least Privilege
- Strong Authentication
- Be Wary of Trust Relationships
- Defense in Depth
- Be Vigilant (watch logs, traffic, etc.)
- Be Educated
- Be Sceptical

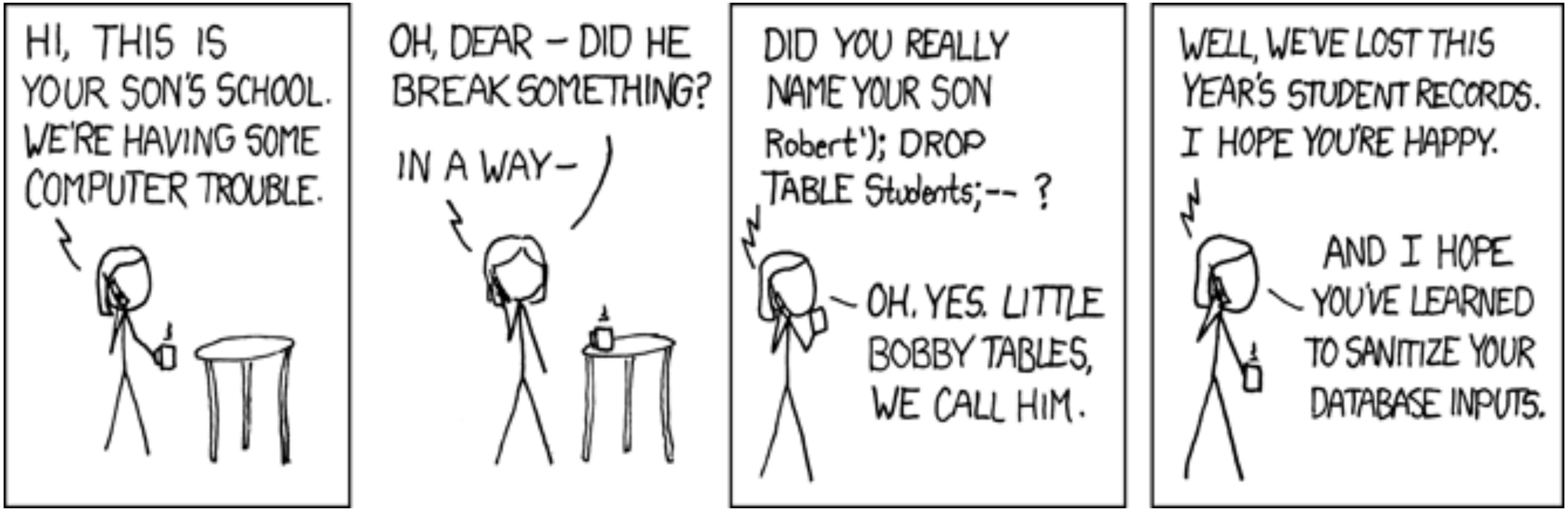
Unix Tools of Defense

- **Secure configuration (STONIX, OpenSCAP)**
  - Least functionality
  - Least privilege
- **File Permissions (PosixACLs)**
- **Configuration Integrity Monitors (AIDE, Tripwire)**
- **One Time Passwords (LinOTP, Mobile-OTP)**
- **Directory Services (LDAP)**
- **Log Aggregation & Correlation (Graylog, Logalyze)**
- **System Auditing (Auditd)**
- **Firewalls (IPTables)**
- **Kerberos**
- **DEP & ASLR (Linux Kernel)**
- **SELinux**
- **Chroot/Virtual Machines (KVM)**
- **Anti-malware (Clam-AV)**
- **Patch Management**
- **Configuration Management (Puppet, cfengine)**
- **IDS/IPS (SNORT)**
- **Backups (Bacula)**
- **Vulnerability Scanning (OpenVAS)**
- **IPSEC**
- **Telnet client**

Resources

- http://www.sans.org/
- http://www.packetstormsecurity.org/
- http://krebsonsecurity.com
- http://www.darkreading.com/
- http://www.schneier.com/
- Blackhat - Defcon
- http://csrc.nist.gov/
- http://www.cisecurity.org/
- Seclists.org
- owasp.org
- shadowserver.org

In Closing


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.